Special Report
CMU/SEI-96-SR-001

Carnegie-Mellon University
Software Engineering Institute

Domain Analysis Workshop Report for the Automated Prompt and Response System Domain

Robert Krut

Nathan Zalman

May 1996

Special Report
CMU/SEI-96-SR-001

May 1996

Domain Analysis Workshop Report for the Automated Prompt and Response System Domain

Robert Krut
Software Engineering Institute

Nathan Zalman
Bell Northern Research, Inc.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

Table of Contents

1 Introduction **1**

2 Domain Analysis **3**
2.1 Feature-Oriented Domain Analysis (FODA) 3
2.2 FODA and the Requirements Analyst 4
2.3 Automated Prompt and Response Systems 5

3 The Context Analysis **7**
3.1 The Structure Diagram 7
3.2 The Context Diagram 9

4 Domain Modeling **12**
4.1 The Features Analysis 12
4.1.1 The Context Features 14
4.1.2 The Operational Features 16
4.1.3 The Representation Features 17
4.2 The Information Analysis 17
4.3 The Operational Analysis 21
4.4 The Domain Dictionary 23

5 Conclusions **24**

References **27**

List of Figures

Figure 2-1	A Summary of the FODA Method	4
Figure 3-1	Structure Diagram for the Domain of Automated Prompt and Response Systems	8
Figure 3-2	Example of an Interface from the Basic Functions Layer	9
Figure 3-3	Context Diagram for the Domain of Automated Prompt and Response Systems	10
Figure 4-1	Partitioning of the FODA Features Model	13
Figure 4-2	Context Features of the Automated Prompt and Response System Domain	14
Figure 4-3	Additional Context Features of the Automated Prompt and Response System Domain	15
Figure 4-4	Operational Features of the Automated Prompt and Response System Domain	16
Figure 4-5	Additional Operational Features of the Automated Prompt and Response System Domain	18
Figure 4-6	Sample 1 of an Information Model for the Automated Prompt and Response System Domain	19
Figure 4-7	Sample 2 of an Information Model for the Automated Prompt and Response System Domain	20
Figure 4-8	Operational Model for the Automated Prompt and Response System Domain	22

Domain Analysis Workshop Report for the Automated Prompt and Response System Domain

Abstract: This report captures the results of the domain analysis tutorial and workshop at Research Triangle Park for Bell Northern Research/Northern Telecom (BNR/NT). Included in this report are brief descriptions of the components of the domain analysis methodology employed and the products developed during the workshop. The information captured within this report will serve as a supplement to the Feature-Oriented Domain Analysis (FODA) tutorial and workshop for the ultimate pilot study domain. The importance of this report is that it provides the future workshop participants with an understanding of the types of products created at a workshop and gives examples of FODA products in a domain familiar to the participants.

1 Introduction

A domain analysis tutorial and workshop were held October 11-13, 1994 at 300 Perimeter Park, Research Triangle Park, North Carolina with employees of Bell Northern Research/ Northern Telecom (BNR/NT) working on the Traffic Operator Position System (TOPS). This tutorial and workshop were part of the first phase of the *Project Plan for TOPS Requirements Management and Capture Pilot Study* [Schnell 94]. This pilot study is designed to demonstrate the use of domain analysis to improve the methodology for capturing and managing customer requirements.

The intent of the tutorial was to

- introduce the attendees to the concept of domain analysis,
- relate this concept to the process of requirements analysis, and
- provide a tutorial on a specific domain analysis methodology with a worked example.

The intent of the workshop was to

- identify a candidate domain within TOPS to apply domain analysis to, and
- attempt to create high level models of the selected domain within TOPS.

This report recaps the information presented during the tutorial and provides the domain specific information generated as part of the workshop discussions. Due to the limited time available, not all facets of the domain analysis methodology were covered as part of workshop discussions. Only those high-level models developed during the discussions are included as part of this report.

The follow-up efforts to the workshop should include a determination as to whether the selected domain is the appropriate domain for the TOPS pilot study. If so, the follow-up effort should continue by extending and refining the captured domain information. If not, a more applicable domain should be selected and analyzed by the methods presented in the tutorial. Both of these options require the commitment of domain analyst and domain experts to the pilot study.

2 Domain Analysis

Domain analysis is "the process of identifying, collecting, organizing, and representing the relevant information in a domain, based upon the study of existing systems and their development histories, knowledge captured from domain experts, underlying theory, and emerging technology within a domain" [Kang 90].

Domain analysis should "carefully bound the domain being considered, consider the ways the systems in the domain are alike (which suggests required characteristics) and the ways they differ (which suggests optional characteristics), organize an understanding of the relationships between the various elements in the domain, and represent this understanding in a useful way" [Nilson 94].

Numerous domain analysis techniques currently exist. Each technique focuses on increasing the understanding of the domain by capturing the information as a model or models. [Nilson 94] discusses six different domain analysis approaches. One such approach is the Feature-Oriented Domain Analysis (FODA), developed at the Software Engineering Institute (SEI).

2.1 Feature-Oriented Domain Analysis (FODA)

The FODA methodology resulted from an in-depth study of other domain analysis approaches. Successful applications of various methodologies pointed towards approaches that focused on the process and products of domain analysis. As a result, the FODA feasibility study [Kang 90] defined a process for domain analysis and established specific products for later use. Such uses of the FODA products include requirements elicitation [Christel 92] and domain design[1] [Peterson 94].

The two basic phases that characterize the FODA process are

1. FODA Context Analysis: defining the extent (or bounds) of a domain for analysis.
2. FODA Domain Modeling: providing a description of the problem space in the domain that is addressed by software.

Figure 2-1 summarizes the inputs, activities, and products of each phase in the FODA process and the relationships between their products. Discussions of each activity and product produced are provided in Sections 3 and 4.

1. Domain design is the process of developing a design model from the products of domain analysis and the knowledge gained from the study of generic architectures and software requirement/design reuse.

Phase	Inputs	Activities	Products*
Context Analysis	operating environments, standards	context analysis	context model [structure diagram, context diagram]
Domain Modeling	features, context model	features analysis	features model [context, operational, representation]
	application domain knowledge	information analysis	information model
	domain technology, context model, features model, information model, requirements	operational analysis	operational model [behavior, functionality]

* As part of the FODA process, domain terminology is captured via a domain dictionary.

Figure 2-1 A Summary of the FODA Method

The feature-oriented concept of FODA is based on the emphasis the method places on identifying prominent or distinctive user-visible features within a class of related software systems. These features are, in essence, the requirements implemented for each of the systems in the domain. Therefore, the intent of this pilot study is to demonstrate the use of FODA to improve the methodology for capturing and managing customer requirements.

2.2 FODA and the Requirements Analyst

The models produced from FODA are used to develop applications in the domain. A key element in the development of these applications is the ability of the requirements analyst to determine the desired requirements for the application. FODA provides models that the requirements analyst can use as a basis for requirements elicitation. For example, the context model can be used by a requirements analyst to determine if the application required by the user is within the domain of an available set of domain products. If the application is within the domain, then the features model can be used by the requirements analyst to negotiate the capabilities of the application with the user.

Typically, a data-flow model has been used as a communication medium between users and developers. However, a data-flow model contains definitions of the internal functions and does not contain the information that the user needs most, which is a description of the external, user-visible aspect of the system. The features model is a better communication medium since it provides this external view that the user can understand.

The information model can be used by a requirements analyst to acquire knowledge about the entities in the domain and their interrelationships. The operational model provides the requirements analyst with an understanding of the domain. The operational model provides the analyst with issues and decisions (other than features) that cause functional differences between the applications. It also captures the rationale for each decision and any constraints or requirements derived from the decisions. From this, the analyst can determine if the operational model can be applied to the user's problems to define the requirements of the application. If the user's problems are all reflected in the features model, then the requirements may be derived from the models by tracing the features, issues, and decisions embedded in the models as parameters. Otherwise, new refinements of the abstract components may have to be made to reflect the functionality and behavior created by the addition of a new requirement or feature.

2.3 Automated Prompt and Response Systems

During the context analysis phase of the workshop, Automated Prompt and Response Systems emerged as a candidate domain. These systems were selected as the domain of interest due to its manageable scope and the in-depth knowledge of the workshop participants.

Automated Prompt and Response Systems are systems which are capable of automating those portions of certain classes of calls which require playing recorded announcements and prompts and collecting subscriber responses. This could include, for example, recording speech or detecting network tones. These systems are embedded in a number of BNR/NT products (e.g., Automatic Alternate Billing System (AABS), Automated DIrectory Assistance System (ADAS), Personalized Automated Response System (PARS™), etc.). Because these systems are common to many BNR/NT systems, they are a likely source of reusable requirements and components.

The following sections provide a brief description of each phase of the FODA methodology. Included are specific examples of the FODA products for the Automated Prompt and Response System Domain. These examples are purposely left as they were generated during the workshop. There are no detailed semantics or syntax in the examples. These relationships would be defined by the resulting tool support employed for the pilot study. A more detailed analysis of prompt and response systems, beyond the elementary analysis done at the workshop, would be necessary to determine the benefits of FODA for improving the methodology for capturing and managing customer requirements.

3 The Context Analysis

The FODA Context Analysis defines the scope of a domain that is likely to yield useful domain products. During the context analysis of a domain, the relationships between the domain of interest and the elements external to it are established and analyzed for variability. An example of the kinds of variability to be accounted for is when applications in the domain have different data requirements and/or operating environments[2]. The results of the context analysis, along with other factors, such as availability of domain expertise, domain data, and project constraints, are used to limit the scope of the domain.

The resulting knowledge from the context analysis provides the domain analysis participants with a common understanding of the

- scope of the domain
- relationship to other domains
- inputs or outputs to or from the domain
- stored data requirements (at a high level) for the domain

The product resulting from the context analysis is the context model. This model includes a structure diagram and a context diagram.

3.1 The Structure Diagram

The FODA Structure Diagram is an informal block diagram in which the domain is placed relative to higher, lower, and peer level domains. The domain under analysis is a part of the higher level domains to which it applies. Lower level domains (or subdomains) are within the scope of the domain under analysis, but are well understood. Any other relevant domains (i.e., peer domains) must also be included in the diagram.

Figure 3-1 provides the structure diagram generated during the workshop for the Automated Prompt and Response System Domain. This diagram captures the layering approach used for generating structure diagrams. The bottom layer (i.e., voice processing platform (VPP), network application vehicle (NAV), voice service node (VSN), interactive voice system (IVS)) represents platforms upon which the basic functions in the domain may be built. For example, the VPP is a hardware/software entity which provides a base for implementing voice processing applications that need to be closely integrated into switch maintenance and call processing. ADAS and central office voice mail are typical examples of services which make use of basic functions and service components implemented on the VPP.

---

2. This variability captured as part of the context analysis may lead to (or be part of) customer requirements.

The second layer (announcements, voice storage, voice recognition, etc.) represents the basic functions, or fundamental capabilities common to most call processing and prompt/response systems. These are implemented using first layer platforms. For instance, voice recognition functionality is exported by hardware and software provided in the first layer.

The third layer represents domains which are the high level building blocks of services. The Automated Prompt and Response System domain is a service component of this type. Queueing systems such as the Traffic Operator Position System Queue Management System (TOPS QMS) are also a domain at this layer.

Finally, the fourth layer consists of call processing applications (most commonly called "services"). Domains at this level contain call flow logic applications which determine the characteristics of calls assigned to them for handling.

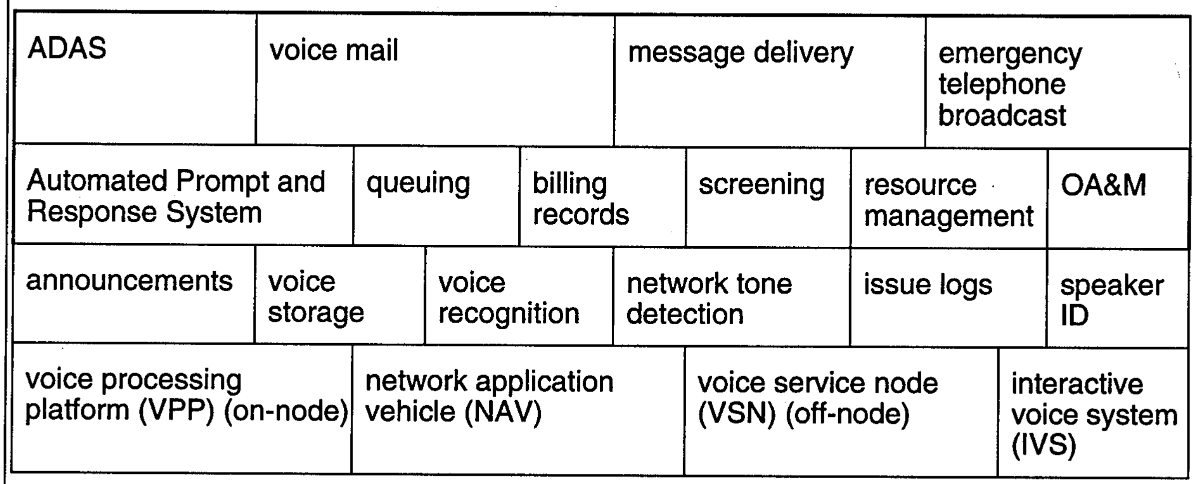

Figure 3-1 Structure Diagram for the Domain of Automated Prompt and Response Systems

The structure diagram layering also aids in understanding the concept of interface layers. An interface layer is defined as the layer on a structure diagram which provides the direct interface between the support packages and the operating environment. Applications above the interface layer typically do not change with operating environment changes whereas applications below change if the operating environment changes. The basic functions layer could be considered an example of an interface layer. This layer would provide platform independence to applications in the service component and services layers.

Voice storage is an example of an interface layer between the automated prompt and response system and the platform upon which the application is built. As shown in Figure 3-2, voice storage would provide an automated prompt and response system, a platform-independent interface that would not change with platform changes. This is the key to understanding platform independence for service components and services.

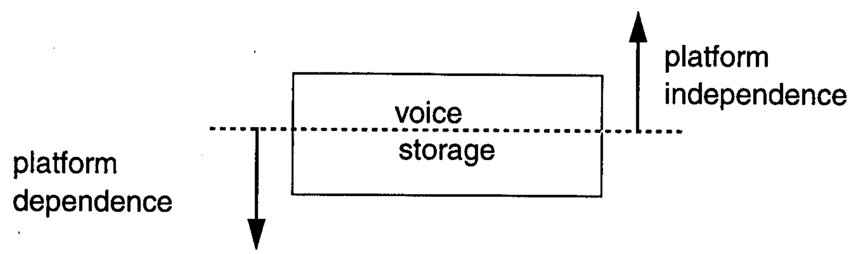

Figure 3-2 Example of an Interface from the Basic Functions Layer

3.2 The Context Diagram

The FODA Context Diagram is a data flow diagram showing data flows between a generalized application within the domain and the other entities and abstractions with which it communicates. One thing that differentiates the use of data flow diagrams in domain analysis from other typical uses is that the variability of the data flows across the domain boundary must be accounted for. This may be done with a set of diagrams, each describing a different context, or with one diagram with the text describing the differences.

Figure 3-3 provides the context diagram generated as part of the workshop for the Automated Prompt and Response System Domain. This drawing captures, as abstractions in most cases, the information provided to and received from the Automated Prompt and Response System Domain. The closed boxes represent the Automated Prompt and Response Systems user as a set of sources and sinks of information. The open-ended box represents the database that the Automated Prompt and Response Systems must interact with. The arrows represent the direction of information flow. Listed below are the principal interactions Automated Prompt and Response Systems engage in and those entities with which they interface.

- *Resource managers* coordinate and control access to Automated Prompt and Response System resources from multiple services.
- *Operations, administration, and maintenance (OA&M)* handles the log and alarm streams from the Automated Prompt and Response System, and provide access to service customization parameters (via administration systems) and system state control (from maintenance systems).
- *File system(s)* represent either internal (i.e., switch resident) or external storage and retrieval systems for digital voice data. This data can take several forms, including voice recognition templates, recorded prompts, recorded subscriber messages, etc.
- *Subscriber(s)* are the end users of the telephone system. This also includes the terminal they use as a vehicle for sending and receiving analog or digital voice data and network tones (via the Dualtone Multifrequency (DTMF) keypad).

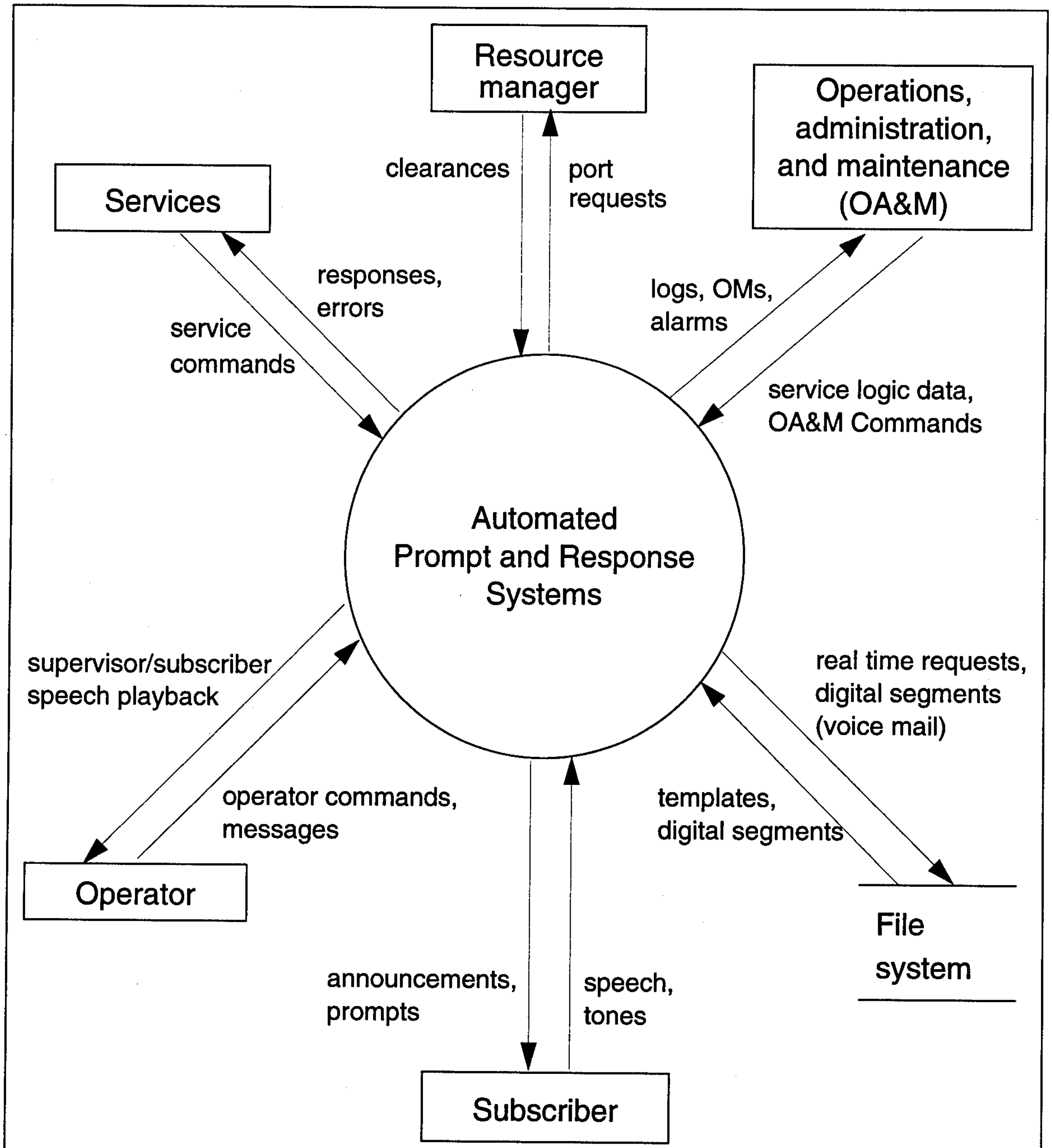

Figure 3-3 Context Diagram for the Domain of Automated Prompt and Response Systems

- *Operator(s)* including the position at which they sit, represent the capability of human interaction with live assistants within the Operator Services environment. Automated Prompt and Response Systems interact with them to provide speech playback, generally of recorded subscriber speech and call arrival tones. There are services, however, which allow the operator to record greetings and other announcements and play them as announcements to subscribers, (e.g., PARS™).

- *Services* is an abstraction for the call flow logic which controls call processing. Its task is to recognize the points-in-call where Automated Prompt and Response System services are required, and to interact with the Automated Prompt and Response System to request it to perform these functions.

4 Domain Modeling

For the domain that is scoped in the context analysis, the FODA Domain Modeling phase identifies and models the commonalities and differences that characterize the applications within the domain. It provides an understanding of the applications in the domain that are addressed by software. By systematically representing (or modeling) the functions, objects, data, and relationships of applications in the domain, domain modeling is used to define what the applications are, what the applications do, and how the applications work. The activities (or analyses) conducted during the domain modeling phase provide an understanding of the

- features of the software in the domain
- standard vocabulary of domain experts
- documentation of the information (entities) embodied in the software
- software requirements via control flow, data flow, and other specification techniques

The FODA Domain Modeling phase produces a domain model which consists of three components. Each component employs a separate analytical technique to model the interrelated components of the domain model (i.e., a features analysis produces a features model, an information analysis produces an information model, and an operational analysis produces an operational model).

The domain modeling process also produces an extensive domain dictionary of terms and/or abbreviations that are used in describing the features and entities in the domain model and a textual description of the features and entities themselves.

4.1 The Features Analysis

The FODA Features Analysis captures a customer's or end user's understanding of the general capabilities of applications in a domain. For a domain, the commonalities and differences among related systems of interest are designated as features[3] and are depicted in the features model. These features, which describe the context of domain applications, the needed operations and their attributes, and representation variations, are important results because the features model generalizes and parameterizes the other models produced in FODA. Therefore, the FODA Features Model partitions features into context, representation, and operational features as seen in Figure 4-1.

---

3. Features are the prominent or distinctive user-visible aspects, qualities, or characteristics of applications in the domain.

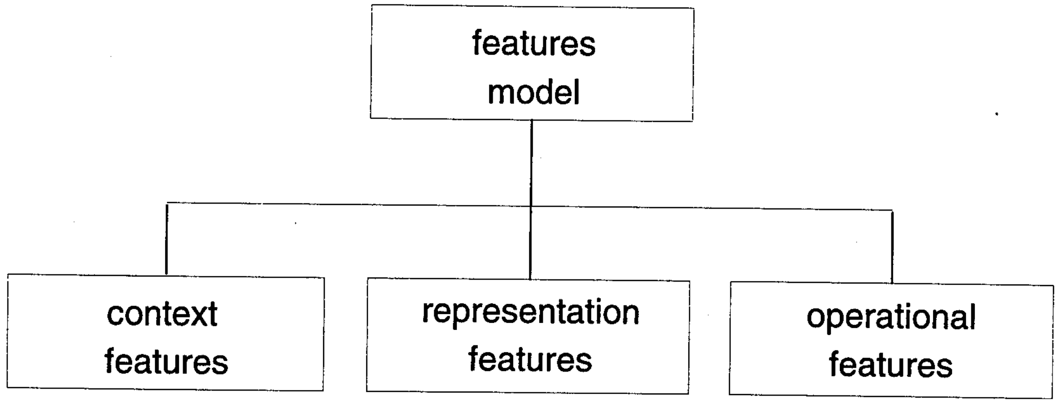

Figure 4-1 Partitioning of the FODA Features Model

Features in the features model may be defined as an alternative, optional, or mandatory[4]. The mandatory features represent the baseline features of an application and the relationships between those features. The alternative and optional features represent the specialization of more general features; that is, they represent what changes are likely to occur over time. With the appropriate features model, one may plan and design for change of a product over time.

The features model is the chief means of communication between the customers and the developers of new applications. The features are meaningful to the end users and can assist the requirements analysts in the derivation of a system specification that will provide the desired capabilities. By providing the end users with a complete and consistent view of the capabilities of applications within the domain, the features model serves as a vehicle by which the end user and requirements analyst communicate system needs.

The following subsections provide a brief description and examples of context, operational, and representation features for the Automated Prompt and Response System Domain. These examples are represented graphically as a set of tree-like diagrams containing a hierarchical decomposition of features. These examples are not complete. They do not contain hierarchical identifiers (such as "consists-of" or "is-a"), identify features as being mandatory, optional, or alternative, or establish connections between the various examples[5].

4. The diagrams produced during the workshop did not, due to time constraints, capture this level of detail. This would be an important step in carrying the work forward.

5. [Krut 93] provides detailed examples of FODA features models as well as a discussion of tool support for previous FODA efforts.

4.1.1 The Context Features

Context features describe the overall mission or usage patterns of an application. Context features also represent such issues as performance requirements, accuracy, and time synchronization that would affect the operations.

Figure 4-2 and Figure 4-3 provide examples of context features generated as part of the workshop for the Automated Prompt and Response System. These features demonstrate different

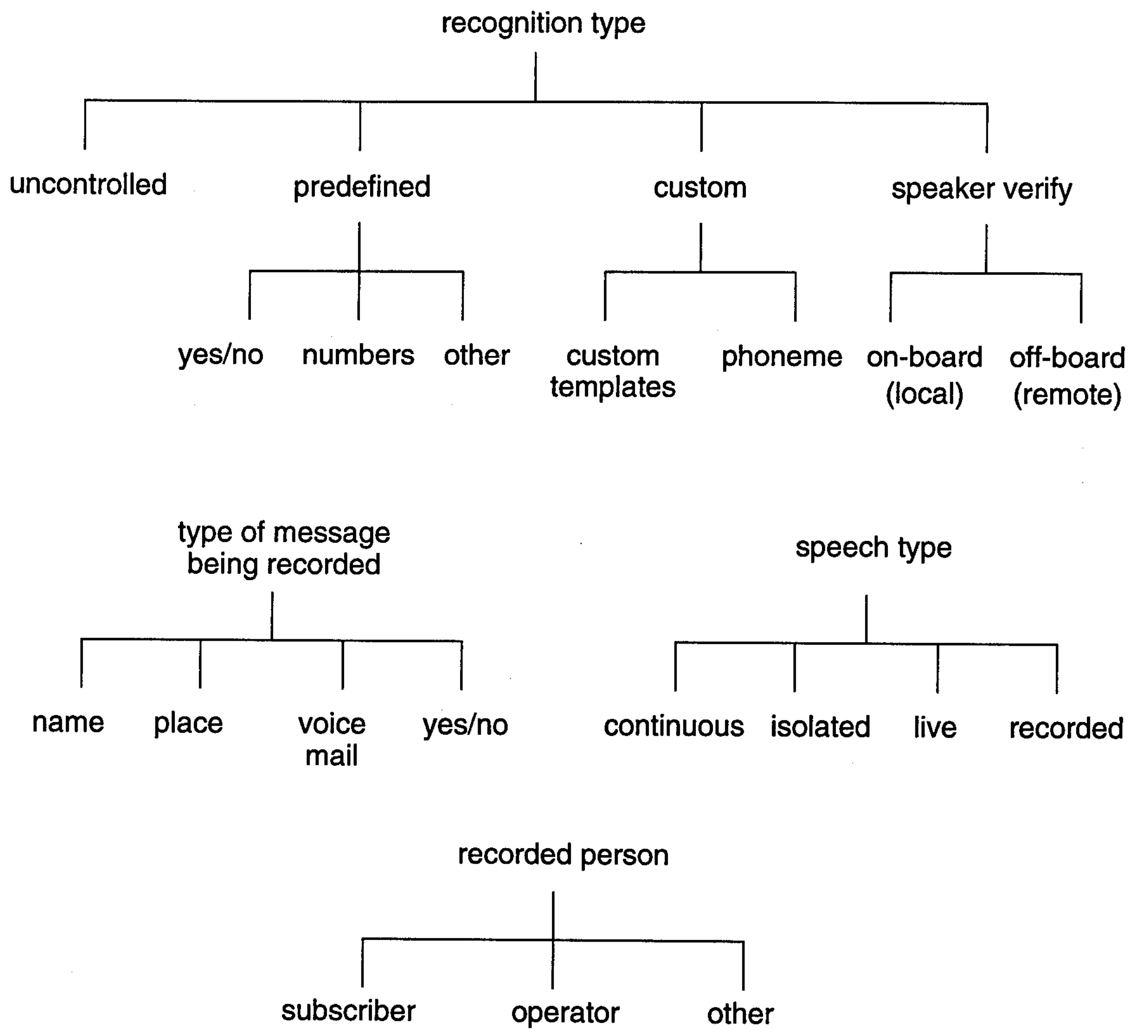

Figure 4-2 Context Features of the Automated Prompt and Response System Domain

ways in which applications within the Automated Prompt and Response System Domain may be used. For example, uncontrolled, predefined, custom, and speaker verify are context features of recognition type. They show that an Automated Prompt and Response System may

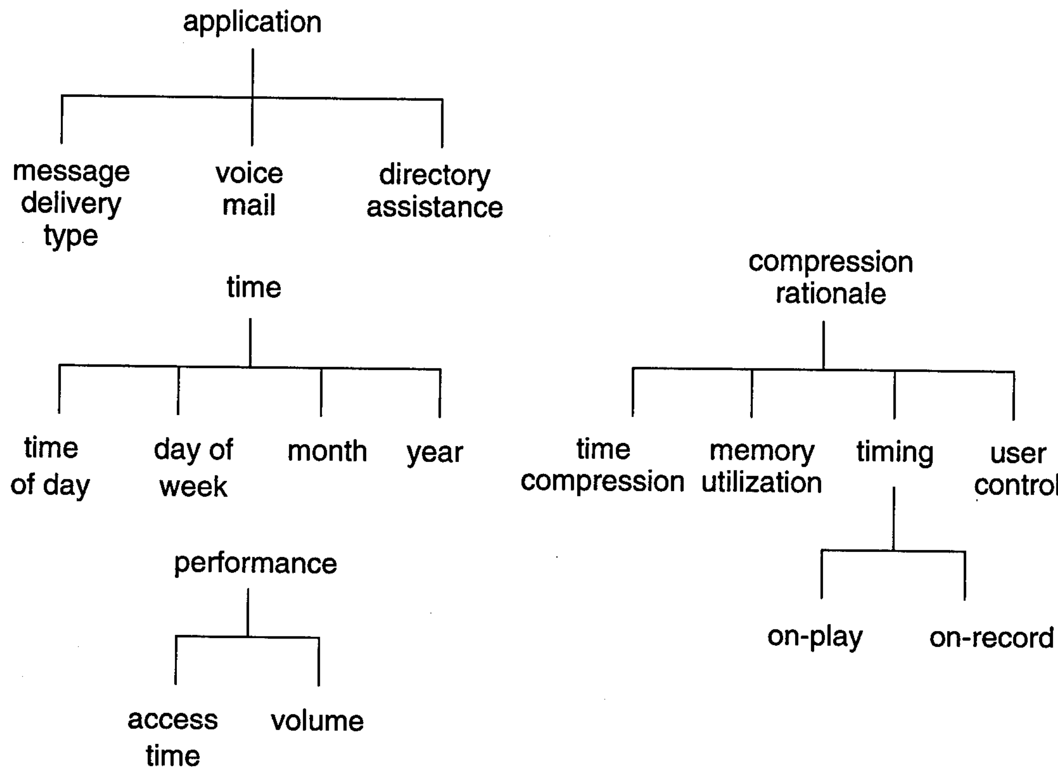

Figure 4-3 Additional Context Features of the Automated Prompt and Response System Domain

implement speech recognition (recognition type) in all or none of the alternatives: arbitrary continuous speech (uncontrolled); limited small vocabulary, speaker-independent (predefined); special vocabulary defined by customer (custom); or speaker-dependent, recognizing the speaker not the words spoken (speaker verify)[6]. Furthermore, within many of the alternatives, refinements are identified, producing a feature variability "tree." This type of feature will easily map to a visible end-user function.

In Figure 4-3, performance is a feature that describes certain characteristics of a system in the context of the overall operating environment (of which Automated Prompt and Response Systems are only a small part). "Access time" and "volume" are only two of the many performance parameters that can be identified and specified as system requirements.

6. Context features support the parameterization of the operational model to establish, depending on the items chosen during the requirements gathering process, very different application capabilities. An example of this type of parameterization is discussed in Section 4.3.

Part of the work of applying a Domain Analysis methodology to the overall "super domains" which constitute BNR/NT's business interests might be to collect and standardize context features of this sort, which are likely to be common to most required applications.

4.1.2 The Operational Features

Operational features are those features that describe the active functions carried out (i.e.,what the application does). These features more closely approximate what is commonly called a "feature" or a "function" within the BNR/NT environment. Figure 4-4 and Figure 4-5 provide examples of operational features generated as part of the workshop for the Automated Prompt and Response Systems.

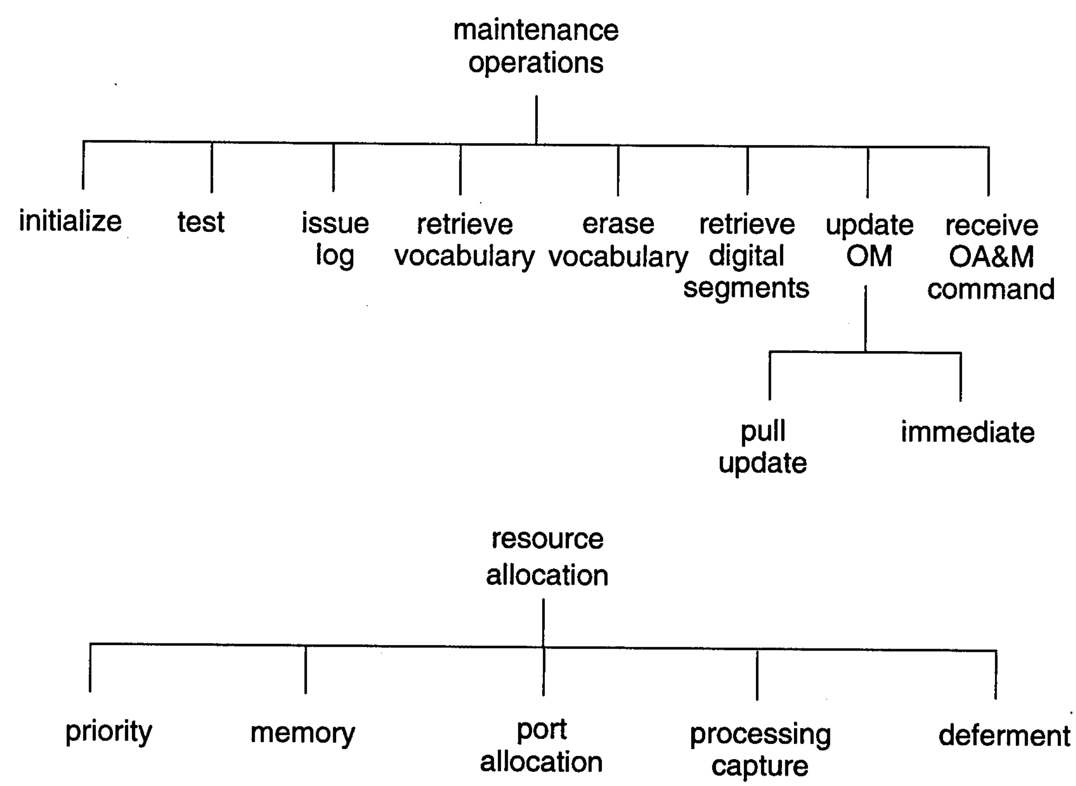

Figure 4-4 Operational Features of the Automated Prompt and Response System Domain

For example, maintenance operations (Figure 4-4) is an operational feature which describes the functional capabilities of maintenance operations or the maintenance services that a system may provide. Maintenance operations consists of one or more of the defined features (i.e., initialize, test, issue log, etc.). Issue log captures the knowledge that a type of maintenance operation is defined for issuing a record of system events. Update operational measurements (Update OM) not only defines that operational measurements can be performed as a maintenance operation but that two options exist for updating the operational measurement: pull update or immediate.

The operational features already exist as part of various applications in the domain. For example, each feature discussed above represents the way in which a maintenance operation is carried out. By capturing these maintenance operations in the operational features models, the requirements analyst and developer can realize what maintenance operations exist and determine whether these operations can be used to fulfill the requirements of a user's application. In essence, the maintenance operations operational features represent a checklist of possible maintenance operations available. If the user requires a maintenance operation not defined on the maintenance operations operational features model, then that operation must be developed from scratch for the new application and incorporated back into the domain model.

4.1.3 The Representation Features

Representation features are those features that describe how information is viewed by a user or produced for another application (i.e., what sort of input and output capabilities are available). No specific examples of representation features were generated at the workshop due to the nature of the domain chosen.

Examples of representation features would be voice, text, or tones for the different user prompts and responses or some type of statistical output (graphs, tables, data streams, etc.) to other users or applications. However, Automated Prompt and Response Systems provide functionality to services, rather than end-user "application" presentations. The nature of the presentation (such as the wording or tone quality of a prompt) could be considered to be under the control of the Services domain (layer four in Figure 3-1).

4.2 The Information Analysis

As part of the FODA Domain Modeling phase, an information analysis captures and defines the domain knowledge and data requirements that are essential for implementing applications in the domain. The domain knowledge is either contextual information which gets lost after the development, or is deeply embedded in the software and is often difficult to trace. The purpose of the information analysis is to represent the domain knowledge explicitly in terms of domain entities and their relationships, and to make them available for the derivation of objects and data definitions during the operational analysis phase.

The product of an information analysis is the information model. The information model may take the form of an entity-relationship (ER) model [Kang 90], a semantic network [Cohen 92], or other representations such as object modeling [Rumbaugh 91].

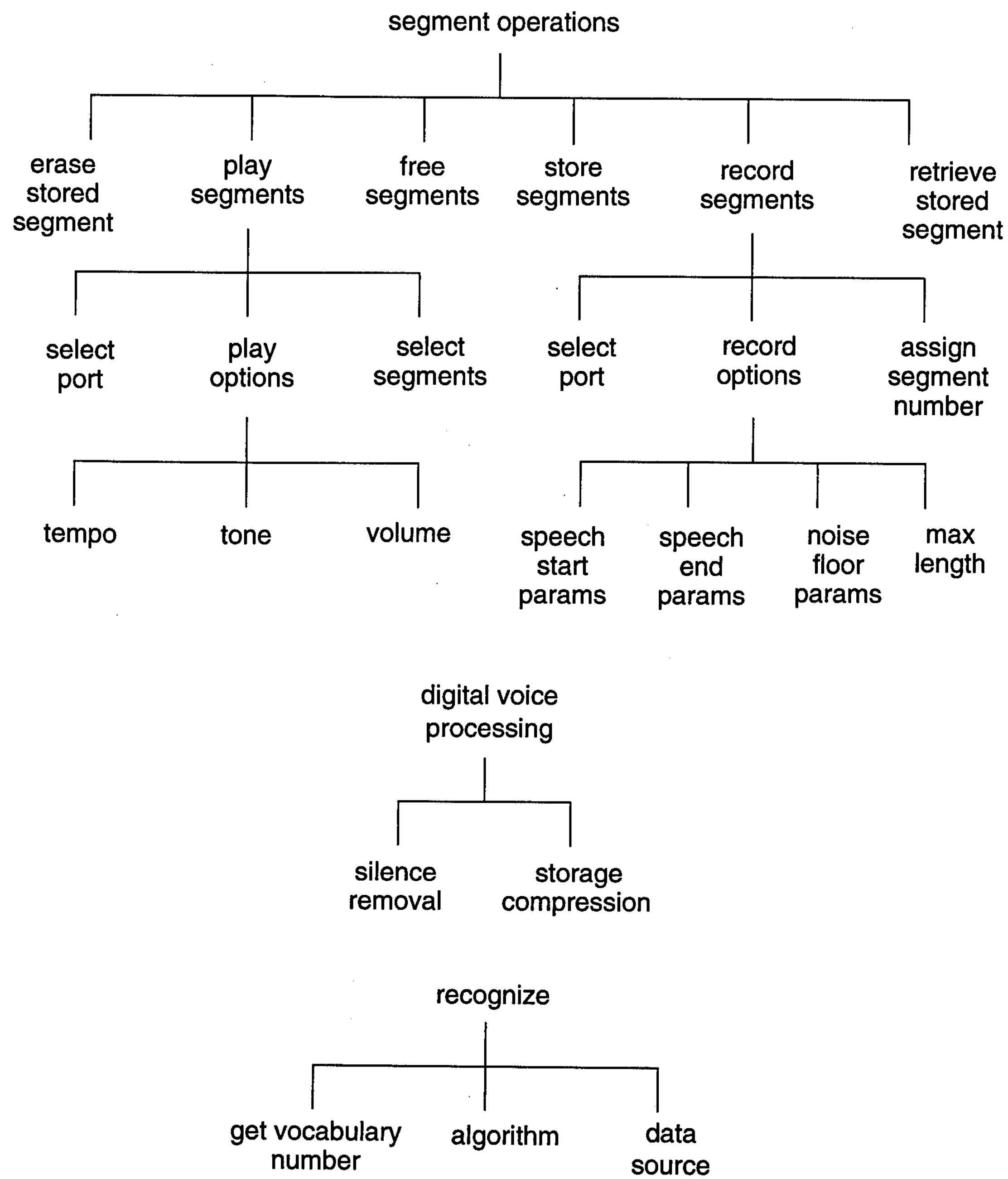

Figure 4-5 Additional Operational Features of the Automated Prompt and Response System Domain

The information model is used primarily by the requirements analyst and the software designer to ensure that the proper data abstractions and decompositions are used in the development of the system. Those who maintain or reuse software need this information. The information model also defines data that is assumed to come from external sources.

Figure 4-6 and Figure 4-7 provide sample information models generated as part of the workshop for the Automated Prompt and Response System Domain. These examples constitute only a small amount of the information which must be captured in the information model. The information in Figure 4-6 and Figure 4-7 represents the following classes of data common to these systems:

- *OA&M measures* are the data associated with many of the maintenance operations described in Figure 4-4, as well as the data flows to the OA&M System as represented in Figure 3-3.
- *Prompts* capture the type of data and its semantic context within a call.
- *Service logic data* corresponds to the parameters to the "Service Commands" data flow in Figure 3-3.
- *Speech data* consists of three sub-types, two of which are then further refined into constituent data elements.

An overall information model for the Automated Prompt and Response System Domain would be built by extending and completing these basic building blocks.

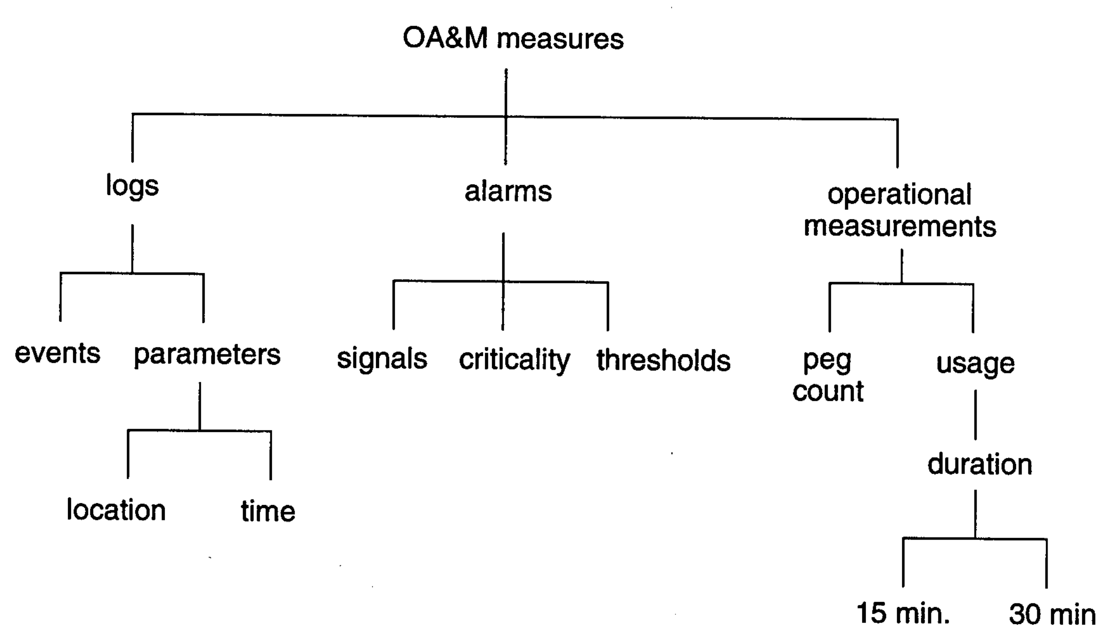

Figure 4-6 Sample 1 of an Information Model for the Automated Prompt and Response System Domain

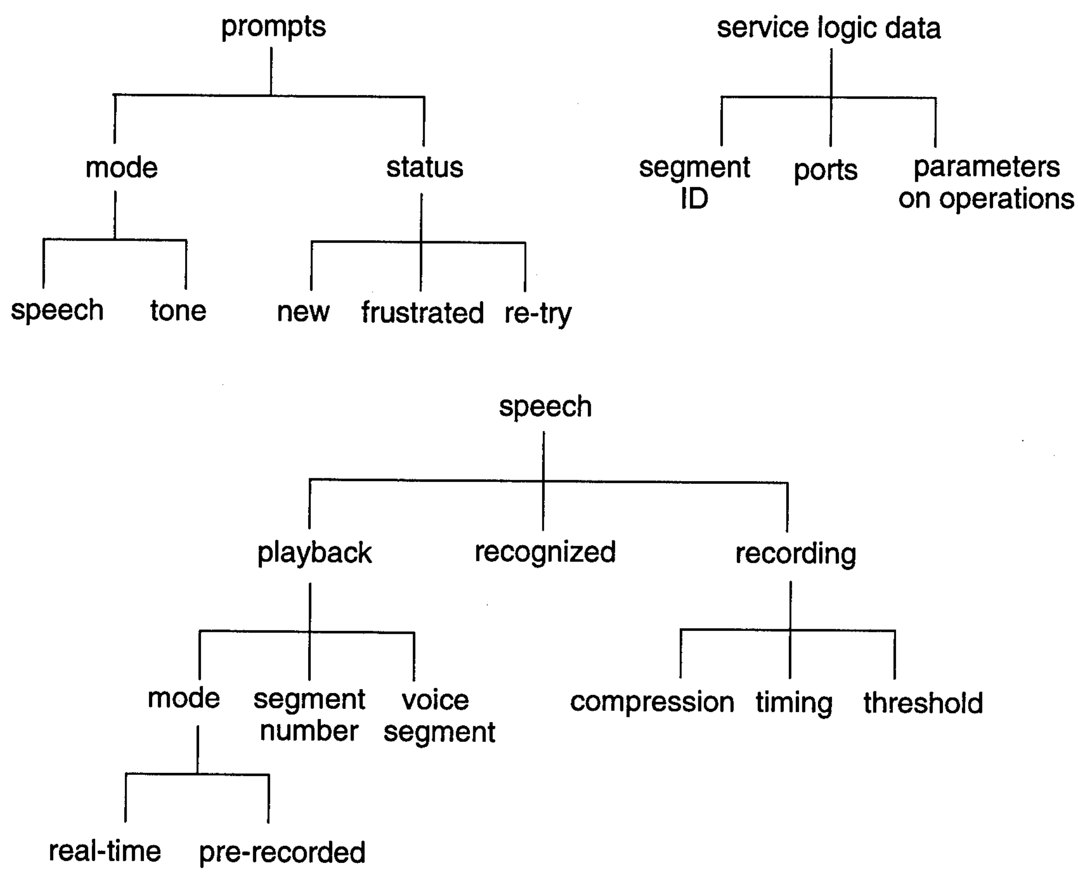

Figure 4-7 Sample 2 of an Information Model for the Automated Prompt and Response System Domain

4.3 The Operational Analysis

The FODA Operational Analysis identifies the control and data flow commonalities and differences of the applications in a domain. This activity abstracts and then structures the common functions found in the domain and the sequencing of those actions into an operational model. Common features and information model entities form the basis for the abstract operational model. Unique features and information model entities complete the operational model. The control and data flow of an individual application can be instantiated or derived from the operational model with the appropriate selection of features.

The operational model represents how the application works. It provides the user with an immediate understanding of applications in the domain. The information may be represented by any means which captures both the behavioral and functional relationship between the objects in the information model and the features in the features model, since activities are driven by features and data flows are driven by the information model.

The operational model is the foundation upon which the software designer begins the process of understanding how to provide the features and make use of the domain entities.

Figure 4-8 provides a sample operational model generated as part of the workshop for the Automated Prompt and Response System Domain. This diagram shows some of the behavioral relationships of the operation to recognize speech types. It also provides an example of how features are used to parameterize the operational model. In this example, the behavioral path traversed would be based on the context feature selected for speech recognition (Section 4.1.1, Figure 4-2). Feature selection will parameterize the operational model, establishing the dynamics of interacting system capabilities.

Within the realm of requirements analysis, the operational model may serve as the basis for constructing application simulations. These simulations can be used to provide a way for the customer to verify that the requirements have been correctly identified (i.e., whether the resulting system will behave as expected).

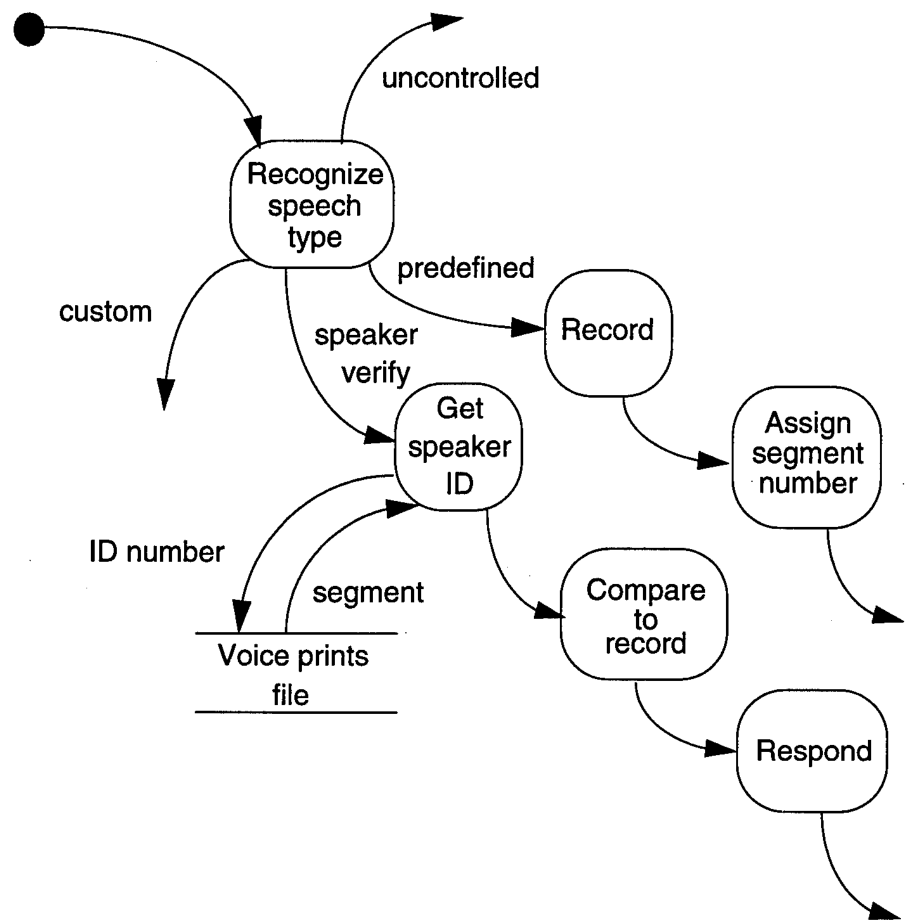

Figure 4-8 Operational Model for the Automated Prompt and Response System Domain

4.4 The Domain Dictionary

The domain dictionary has been found to be one of the most useful products of a domain analysis. The dictionary helps to alleviate a great deal of miscommunication by providing the domain information users with

- a central location to look for terms and abbreviations that are completely new to them
- definitions of terms that are used differently or in a very specific way within the domain

Definitions for the domain dictionary come from such sources as requirements documents, specifications, and discussions with application and domain experts. In addition to the definition of the terms and/or abbreviations, the domain dictionary would list synonyms, source(s) of the definition, the hierarchical abstraction and decomposition of the definition, the feature type (i.e., mandatory, optional, or alternative) if the terms represents a feature, and any additional information deemed necessary to completely understand the term and/or abbreviation.

No attempt was made to capture any definitions for an Automated Prompt and Response System Domain Dictionary during the workshop. This effort extends beyond the development of small model examples seen throughout this report. However, if the analysis of the Automated Prompt and Response System Domain continues beyond the workshop, the existing information must be captured in the domain dictionary and evolve as the domain models are developed.

5 Conclusions

The natural conclusion to a workshop report would tend to prompt questions and comments such as:

> *Is this the domain of interest for the pilot study? If so, then the analysis must begin with a review of the material captured and build upon this material until the domain model is completed. This work would include the development of a detailed domain dictionary.*
>
> *What support tool is going to be used to create and maintain the models? Is there going to be any attempt to extend the captured information into a simulation or prototype?*

Since the workshop, several of these questions have been addressed. The following lists the direction that the pilot study will take beyond the work captured for the Automated Prompt and Response System Domain.

Is this the domain of interest for the pilot study?

> Due to staffing and time constraints, the pilot study will not pursue the Automated Prompt and Response System Domain as its target domain. Another target domain has been identified within BNR, and detailed planning of the modeling effort is well under way.
>
> It is intended that this document be used as a tool for presenting FODA Domain Modeling concepts to the domain experts who will participate in the second round of the pilot study.

Should a domain dictionary be created?

> This work will not be completed within the Automated Prompt and Response System domain. Many of the representation issues which need to be addressed in order to complete a domain dictionary, though addressed in part through some of the work done in item (2) above, will be deferred until after tool selection is completed.

What support tool is going to be used to capture the models?

> A variety of tools are being investigated. Some preliminary work with model representation using Smart Elements (SE), a product of Neuron Data Corporation, has been done. This work points to SE as a promising tool for generating model applications, though not necessarily for generating the model database itself without extensive development effort.
>
> Future work in this area should concentrate on identifying tools which have already been developed and are currently in use within industry for model capture.

As a result of these decisions, the work on the Automated Prompt and Response System Domain will stop, but the information captured within this report will serve as a supplement to the FODA tutorial and workshop for the ultimate pilot study domain. The importance of this report is that it provides the future workshop participants with an understanding of the types of products created at a workshop and gives examples of FODA products in a domain familiar to the participants.

References

[Christel 92] Christel, Michael G.; Kang, Kyo C. *Issues in Requirements Elicitation* (CMU/SEI-92-TR-12, ADA258932). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1992.

[Cohen 92] Cohen, Sholom G.; Stanley Jr., Jay L.; Peterson, A. Spencer; & Krut Jr., Robert W. *Application of Feature-Oriented Domain Analysis to the Army Movement Control Domain* (CMU/SEI-91-TR-28, ADA256590). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1992.

[Kang 90] Kang, Kyo C.; Cohen, Sholom G.; Hess, James A.; Novak, William E.; & Peterson, A. Spencer. *Feature-Oriented Domain Analysis (FODA) Feasibility Study* (CMU/SEI-90-TR-21, ADA235785). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990.

[Krut 93] Krut Jr., Robert W. *Integrating 001 Tool Support into the Feature-Oriented Domain Analysis Methodology* (CMU/SEI-93-TR-11). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

[Nilson 94] Nilson, Roslyn; Kogut, Paul; & Jackelen, George. *Component Provider's and Tool Developer's Handbook Central Archive for Reusable Defense Software (CARDS).* (STARS-VC-B017/001/00). Reston, VA: Unisys Corporation, 1994.

[Peterson 94] Peterson, A. Spencer; Stanley Jr., Jay L. *Mapping a Domain Model and Architecture to a Generic Design* (CMU/SEI-94-TR-08, ADA 283747). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1994.

[Rumbaugh 91] Rumbaugh, James, et al. *Object-Oriented Modeling and Design.* Englewood Cliffs, NJ: Prentice Hall, 1991.

[Schnell 94] Schnell, Karen. *Project Plan for TOPS Requirements Management & Capture Pilot Study.* Research Triangle Park, NC: Bell-Northern Research, Inc., 7 October 1994.

UNLIMITED, UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-96-SR-001	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute 6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213	7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office 8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-95-C-0003
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO: 63756E PROJECT NO.: N/A TASK NO: N/A WORK UNIT NO.: N/A

11. TITLE (Include Security Classification)
Domain Analysis Workshop Report for the Automated Prompt and Response System Domain

12. PERSONAL AUTHOR(S)
Robert Krut, Nathan Zalman

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	May 1996	28

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
FIELD	GROUP	SUB. GR.	domain analysis, automated prompt and response system domain, feature-oriented domain analysis, FODA, Software Engineering Institute, SEI, Bell Northern Research, BNR

19. ABSTRACT (continue on reverse if necessary and identify by block number)

This report captures the results of the domain analysis tutorial and workshop at Research Triangle Park for Bell Northern Research/Northern Telecom (BNR/NT). Included in this report are brief descriptions of the components of the domain analysis methodology employed and the products developed during the workshop. The information captured within this report will serve as a supplement to the Feature-Oriented Domain Analysis (FODA) tutorial and workshop for the ultimate pilot study domain. The importance of this report is that it provides the future workshop participants with an understanding of the types of products created at a workshop and gives examples of FODA products in a domain familiar to the participants.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT☐ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF	22b. TELEPHONE NUMBER (include area code) (412) 268-7631 22c. OFFICE SYMBOL ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19